Reconfigurable VLSI Architectures for Evolvable Hardware: From Experimental Field Programmable Transistor Arrays to Evolution-Oriented Chips

Adrian Stoica, Ricardo Zebulum, Didier Keymeulen, Raoul Tawel, Taher Daud, and Anil Thakoor

***Abstract*—Evolvable hardware (EHW) addresses on-chip adaptation and self-configuration through evolutionary algorithms. Current programmable devices, in particular the analog ones, lack evolution-oriented characteristics. This paper proposes an evolution-oriented field programmable transistor array (FPTA), reconfigurable at transistor level. The FPTA allows evolutionary experiments with reconfiguration at various levels of granularity. Experiments in SPICE simulations and directly on a reconfigurable FPTA chip demonstrate how the evolutionary approach can be used to automatically synthesize a variety of analog and digital circuits.**

***Index Terms*—Adaptive computing, evolvable hardware, field-programmable transistor arrays (FPTAs), genetic algorithms, reconfigurable VLSI architectures.**

I. Introduction

The idea behind evolutionary circuit synthesis/design and evolvable hardware (EHW) is to employ a genetic search/optimization algorithm that operates in the space of all possible circuits and determines solution circuits that satisfy imposed specifications, including target functional response, size, speed, power, etc.

Currently, the search for a circuit solution can be performed using software simulations [1] or directly in hardware on reconfigurable chips [2], [17]. However, software simulations take too long for practical purposes, since the simulation time for one circuit is multiplied by the large number of evaluations required by evolutionary algorithms. In addition the resulting circuit may not be easily implemented in hardware, unless implementation constraints are imposed during evolution. Hardware evaluations can reduce by orders of magnitude the time to get the response of a candidate circuit, potentially reducing the evolution time from days to seconds [3]. Hardware evaluations commonly use commercial reconfigurable devices, such as field programmable gate arrays (FPGAs) [4] or field programmable analog arrays (FPAAs) [3]. These devices, designed for several applications other than EHW, lack evolution-oriented features, and, in particular the analog ones, are suboptimal for EHW applications.

This paper addresses devices specifically targeted for EHW, in particular those allowing analog processing. It proposes an architecture on which both analog and digital circuits can be synthesized by evolutionary means. The paper is organized as follows. Section II provides background on circuit synthesis using evolutionary algorithms. Section III reviews efforts toward evolution-oriented reconfigurable architectures (EORA), focusing on programmable analog devices. Section IV introduces the FPTA and Section V illustrates evolutionary experiments performed on this device. Following these, portability, mixtrinsic evolution issues are discussed in Section VI followed by conclusions in Section VII.

Manuscript received February 17, 2000; revised July 18, 2000. This work was performed at the Center for Integrated Space Microsystems, Jet Propulsion Laboratory, California Institute of Technology. This work was supported by the Defense Advanced Research Projects Agency and by the National Aeronautics and Space Administration.

The authors are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 USA (e-mail: adrian.stoica@jpl.nasa.gov).

Publisher Item Identifier S 1063-8210(01)00802-2.

II. Background

The genetic search in EHW is tightly coupled with a coded representation that associates each circuit to a "genetic code" or chromosome. The simplest representation of a chromosome is a binary string, a succession of 0s and 1s that encodes a circuit. The main steps of evolutionary synthesis are illustrated in Fig. 1.

First, a population of chromosomes is randomly generated. The chromosomes are converted into circuit models for evaluation in SW, in which case the evolution is called extrinsic, or into control bitstrings downloaded to programmable hardware. The latter is referred to as intrinsic EHW. Circuit responses are compared against specifications, and individuals are ranked based on how close they come to satisfying them. In preparation for a new iteration, a new population of individuals is generated from the pool of best individuals in the previous generation. This is subject to a probabilistic selection of individuals from a best individuals pool, followed by two operations: random swapping of parts of their chromosomes, the *crossover* operation, and random flipping of chromosome bits, the *mutation* operation. The process is repeated for several generations, resulting in increasingly better individuals [1]. Randomness helps to avoid getting trapped in local optima. Monotonic convergence (in a loose Pareto sense) can be forced by unaltered transference to the next generation of the best individual from the previous generation. There is no theoretical guarantee that the global optimum will be reached in a useful amount of time; however, the evolutionary/genetic search is considered by many to be the best choice for very large, highly unknown search spaces. The search process is usually stopped after a number of generations or when closeness to the target response has reached a sufficient degree. One or several solutions may be found among the individuals of the last generation.

Evolution can either be online or offline. Evolution is online when the circuit actively changes the behavior of a functional system such as a robot operating in a target environment.

III. Reconfigurable Architectures for EHW Experiments

A. Toward Evolution-Oriented Reconfigurable Architectures

This discussion is mainly in the context of devices supporting evolution of analog circuits. To best support EHW one must consider several aspects for EORA. The granularity of the programmable chip is an important feature. A first limitation of FPAAs is their coarse granularity, basically the operational amplifier level. For FPGAs the finest granularity is at the gate level, which may be sufficient for evolution of digital circuits. However, using FPGAs to evolve analog circuits, as shown possible in [2], is not an efficient technique for obtaining analog functionality.

From the EHW perspective, it is interesting to have *programmable granularity*, allowing the sampling of novel architectures together with the possibility of implementing standard ones. The optimal choice of elementary block-type and granularity is task dependent. At least for experimental work in EHW, it appears a good choice to build reconfigurable hardware based on elements of the lowest level of granularity. Virtual higher level building blocks can be considered by imposing programming constraints. An example would entail forcing groups of elementary cells to act as a whole, e.g., by freezing certain parts of their configuration bitstrings that describe say a NAND gate. Ideally, the "virtual blocks" for evolution should be automatically defined/clustered during evolution [1].

EORA should be *transparent architectures*, allowing the analysis and simulation of the evolved circuits. They should also be robust enough not to be damaged by any bitstring configuration existent in the search space, potentially sampled by evolution. Finally, EORA should allow evolution of both analog and digital functions.

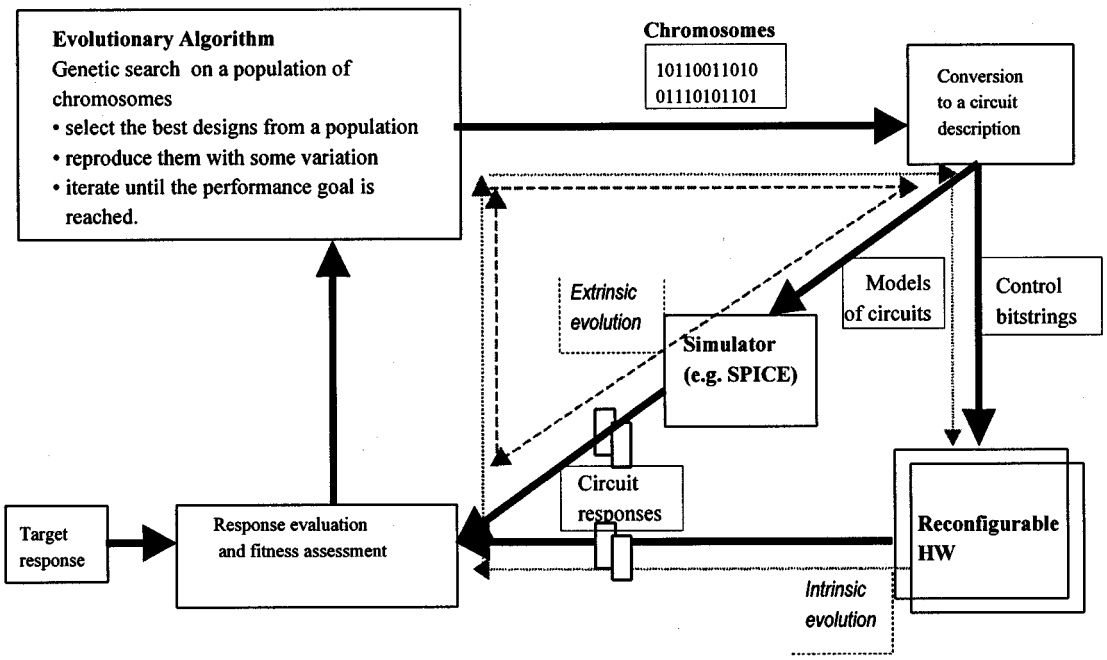


Fig. 1. Main steps for the evolutionary synthesis of electronic circuits.

B. Overview of Several Reconfigurable Devices Used in Evolutionary Experiments

Totally Reconfigurable Analogue Hardware (TRAC) (Zetex) [5] is a coarse grained FPAA, consisting of 20 operational amplifiers OpAmp laid out as two parallel sets of ten interconnectable amplifiers. Each amplifier has various components around it that may be switched in and out of circuit by programming a shift-register to give one of these distinct operations: off; pass; invert; exponentiation; logarithm; rectify; and sum of two inputs. Its main limitation in EHW is that evolution is constrained to the arrangement of high level conventional building blocks to compose the final topology, not being able to arrive at any novel design.

MPAA020 Field Programmable Analog Array (Motorola) [6] is a coarse grained FPAA with analog resources distributed along 20 almost identical cells, each of which includes OpAmp, comparators, capacitors, programmable switches, and SRAM. It was used as a platform for the evolution of filters, oscillators, amplifiers, and rectifiers [7]. This chip provides more resources to evolution since programmable capacitors and resistors, implemented through the switched capacitor effect, are included on chip. However, since the basic building blocks are fixed to OpAmp topologies, there is little room for evolution to arrive at novel designs. Another limitation is its opaque architecture, preventing simulation/analysis of evolved circuits.

Palmo (Univ. Edinburgh) [8] is a BiCMOS coarse-grained FPAA in which digital pulses are used to represent discrete analog signals. The chip architecture consists of an array of programmable cells that perform the functionality of integrators.

EM Evolvable Motherboard (Univ. Sussex) [9] is a research tool designed to access low level granularity. This programmable board allows discrete devices to be plugged in as basic circuit elements. The board is organized as a matrix of components, where digitally controlled analog switches allow row/column interconnection. It is used to evolve inverters, amplifiers, and oscillators, using bipolar transistors.

PAC Programmable Analog Circuits (Lattice) [10] is a coarse-grained programmable analog circuit intended for applications in filtering, summing/differentiating, gain/attenuation, and conversion. The PAC cell includes three instrumentation transconductance amplifiers, two of them with programmable transconductances. This architecture also provides an on-chip programmable capacitor, consisting of a parallel arrangement of seven capacitors, each in series with an E^2CMOS switch. A drawback from EHW perspective is the fact that it is programmed through nonvolatile E^2CMOS memory cells, which limit the number of program/erase cycles to around 100 000, a limitation for evolutionary experiments.

C. Summary and Discussion

Table I summarizes the main features of the surveyed devices. The devices have limitations from the point of view of their suitability in implementing EORA. The main problem of the commercial devices (Xilinx, TRAC, MPAA020, and Lattice) is their coarse granularity and, in some cases, the nontransparent architecture. The academic research tools (EM and Palmo) are board level solutions. EM has more flexibility since any component can be added by human plug-in; this offers enhanced capabilities but it also means that humans can bias the search for a solution. Palmo is coarse grained, which limits the evolution search space.

IV. Field Programmable Transistor Array

The FPTA was proposed as a flexible, versatile platform for EHW experiments and developed as an intermediate step toward a stand-alone system on a chip [3]. As a first step toward EORA, the FPTA is a concept design for hardware reconfigurable at transistor level. Since both analog and digital CMOS circuits ultimately rely on functions implemented with transistors, the FPTA appears as a versatile platform for the synthesis of both analog and digital (and mixed-signal) circuits. The architecture is cellular and has similarities with other cellular architectures; e.g., those of Xilinx XC6200 family FPGAs or cellular neural networks. One key distinguishing characteristic relates to the definition of the elementary cell. The architecture is largely a "sea of transistors" interconnected by other transistors that act as signal passing devices. In order to control the opening of the switch, a HIGH or LOW control signal is determined by one bit in the chromosome. The actual voltage levels associated with the HIGH

TABLE I
THE MAIN FEATURES OF THE SURVEYED PROGRAMMABLE DEVICES (NA—INFORMATION NOT AVAILABLE)

Feature /Device	TRAC	MPAA020	PALMO	EM	Lattice
Granularity	coarse	coarse	coarse	fine	coarse
Protection	NA	software tools	software/ hardware	switches parasitics	NA
Circuit Download	parallel port	serial port	serial port	ISA bus	serial port
Proprietary Information	NA	Yes	NA	No	Yes
Search Space	NA	~2^{300}/cell	NA	10^{420}	NA
Technology	CMOS	CMOS	BiCMOS	Board level	CMOS

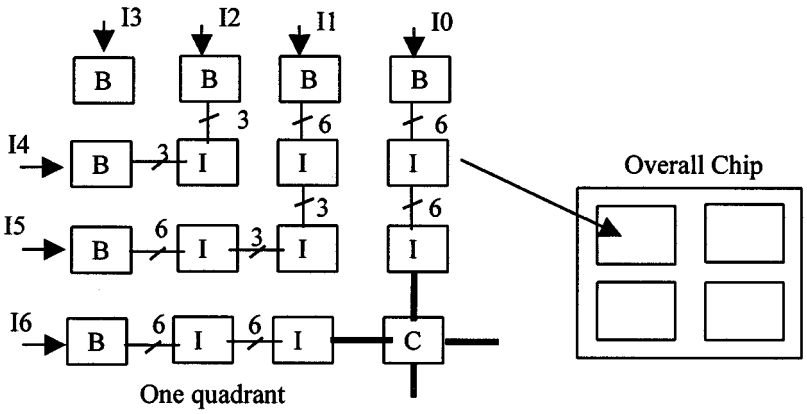


Fig. 2. One quadrant of the FPTA chip with boundary (B), intermediate (I), and central (C) cells.

and LOW are imposed by external analog input signals and are the same for all the switches in that particular state. Continuous voltage control allows the switches to be only partly opened; this leads to intermediate resistance values found useful for artificial evolution (gray-level switches [3]). Otherwise said, for all switches, "1" means ON, but, depending on a control voltage, the resistance values may be 50 Ohms, 100 Ohms, or some other value.

The global FPTA architecture and its cell topology are described in the following.

A. Global Architecture of the FPTA Chip

Fig. 2 shows the block diagram of prototype chip organized as an 8×8 matrix of cells. Due to the symmetry of the chip, only the first quadrant is shown in the figure. Each cell can be configured at the transistor level to perform different analog and digital functions.

The cells are divided into three different categories according to their relative position in the array: boundary cells (B), intermediate cells (I), and central cells (C). There are 28 boundary cells, each of which receives one external input. A total of 24 external inputs (I0 to I23) can be applied to the chip via 24 pins. There are 32 intermediate cells and 4 central cells. The boundary and intermediate cells are connected through six different programmable switches. A more flexible interconnection pattern is allowed for the central cells. Each central cell can connect to its West, East, South, and North neighbors through analog multiplexers. The multiplexed connections are represented in the figure by the four thick lines coming to the central cell. The central cells are the only cells in the chip with capacitance resources.

In order to allow partial reconfiguration of the chip, each particular cell can be independently addressed and reprogrammed through a simple decoding mechanism. Around 30 bits are needed to program each cell, allowing a cell reconfiguration time of a few micro-seconds.

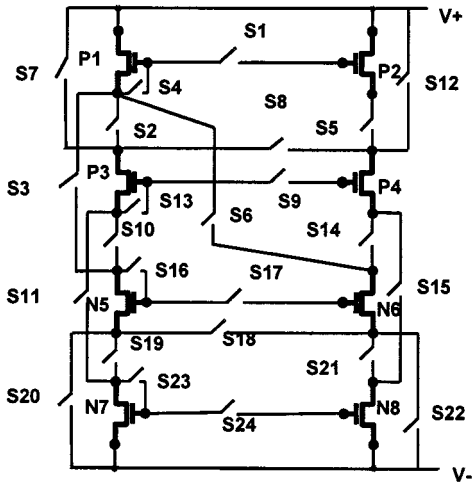


Fig. 3. Schematic of an FPTA cell consisting of 8 transistors and 24 switches.

B. FPTA Cell

The cell is an array of transistors interconnected by programmable switches. The status of the switches (ON or OFF) determines a circuit topology and consequently a specific response. Thus, the topology can be considered as a function of switch states, and can be represented by a binary sequence, such as "1011. . .," where a "1" is associated to a switch turned ON and a "0" to a switch turned OFF. Fig. 3 illustrates the schematic of the FPTA cell consisting of 8 transistors and 24 programmable switches.

In this implementation, transistors P1–P4 are PMOS and N5–N8 are NMOS, and the switch-based connections are in sufficient number to

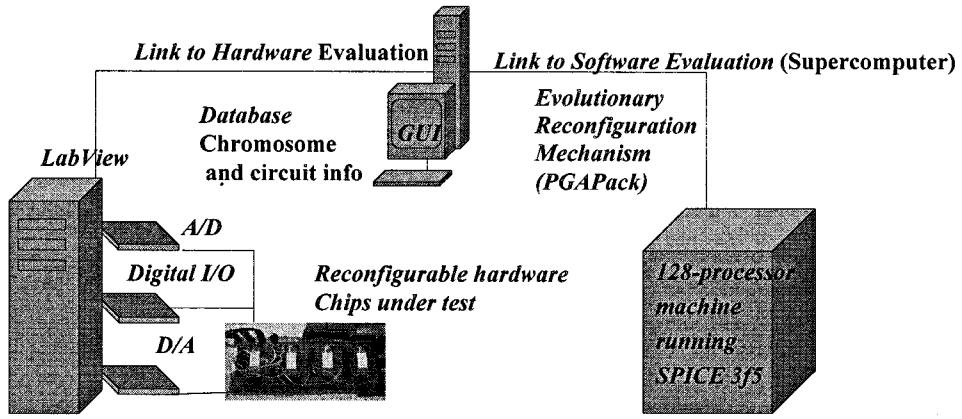


Fig. 4. EHW testbed 5 allows evolutions in SPICE simulation as well as directly on the chip.

allow a majority of meaningful topologies for a given transistor arrangement and yet less than the total number of possible connections.

The FPTA cell was fabricated in 0.5 μm CMOS technology. A test chip contains two FPTA cells with variations of transistor parameters. The chip has allowed circuits obtained through evolution in simulations to be validated by downloading and evaluating their performance in hardware. More importantly, it has offered a platform to study hardware issues related to evolution, enabling intrinsic evolution at an accelerated pace. Over 100 times speed-up is possible in comparison to the simulation on a supercomputer ($\sim$5 s compared to $\sim$20 min on a 128-processor parallel machine, for the evolution of a Gaussian circuit described later).

Reconfiguration at transistor level allows definition of building blocks or subcircuits at a variety of levels of granularity. At the lowest level one can configure subcircuits such as current mirrors and differential pairs using only one cell, while more complex blocks, such as logical gates and OpAmps, can also be easily configured with one or two cells. The level of accessible granularity can be set by the designer, who can either let the evolutionary process manipulate the cell at the transistor level or freeze the cells architecture to well known high level analog building blocks and let evolution manipulate them. In the former approach, one can expect evolution to come with the building blocks that are most suitable for the particular application.

The mapping of conventional design blocks illustrates the representation capability of the architecture. Two examples, for a transconductance amplifier and logical NAND/AND gates, are detailed in [11].

V. Evolution on FPTA

A. Evolvable Hardware Testbed

An EHW testbed, shown in Fig. 4, was developed to support both SW and HW evaluations (extrinsic/intrinsic). The SW resources rely on an SGI 128-processor parallel machine running multiple copies of SPICE. The HW resources are built around National Instruments LabView, associated data acquisition boards, signal generators, and other equipment [3].

We developed an SW tool called EHWPack, which incorporates the public domain Parallel Genetic Algorithm package PGAPack [12] as the genetic engine and SPICE 3F5 as the circuit simulator. An interface code links the GA with the simulator where potential designs are evaluated, while a graphical user interface (GUI) allows easy problem formulation and visualization of results. At each generation the GA produces a new population of binary chromosomes, which get converted into netlists that describe candidate circuit designs, and are further simulated by SPICE. In parallel with the circuit simulations, the chromosomes may also be downloaded on a test board with four FPTA test chips, and the circuit response read back through a TCP/IP connection. Thus evolution can be extrinsic, intrinsic, or a combination of the two (mixtrinsic), to which we refer in Section VI.

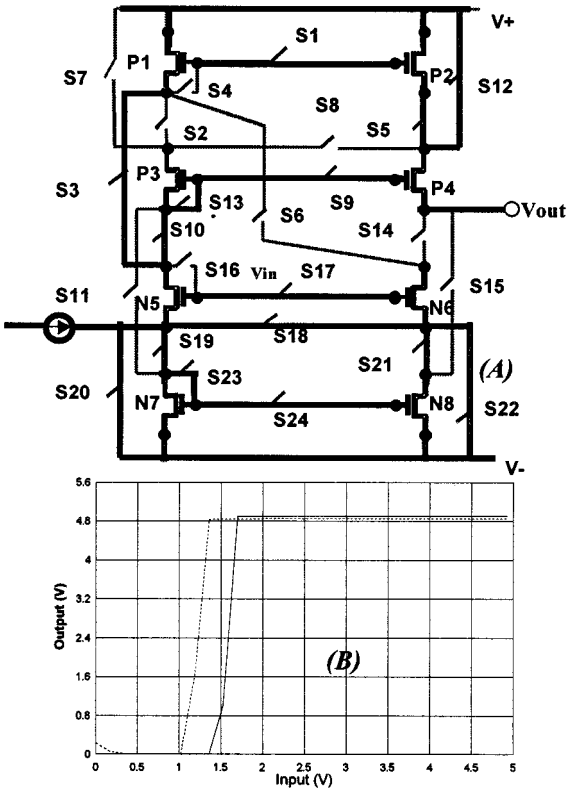


Fig. 5. (a) Schematic of an evolved amplifier using the FPTA model and (b) comparison of the dc characteristics displayed by the simulated and implemented versions of the amplifier.

The following describes experiments on the evolutionary synthesis of analog building blocks. The same evolutionary method (GA) was employed in all experiments using different evaluation functions according to the problem specification.

B. Synthesis of Analog Signal Processing Circuits

1) Amplifier: The objective of this experiment was to synthesize a circuit with the dc transfer characteristic typical of an amplifier. Fig. 5 depicts the schematic of the evolved circuit, together with the dc transfer responses achieved in the FPTA implementation and in simulation.

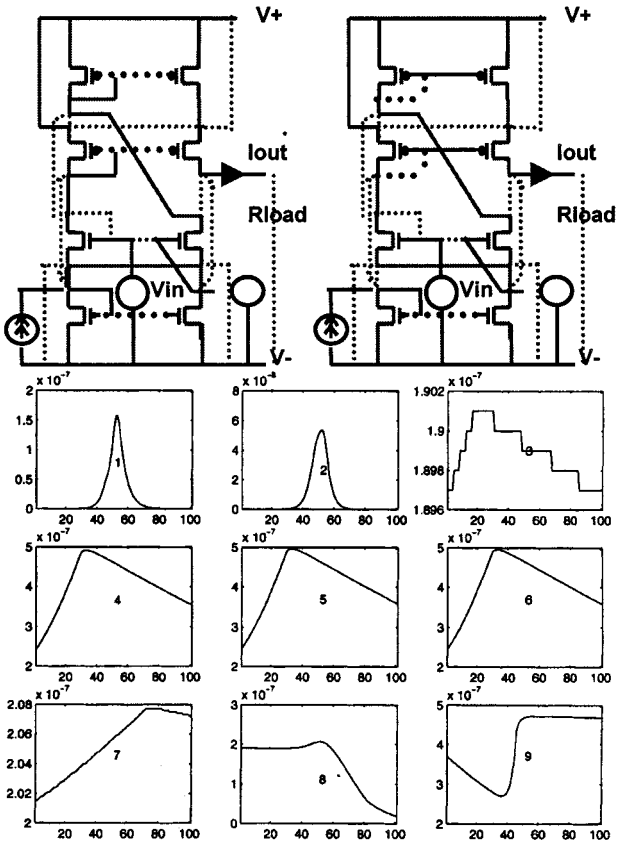
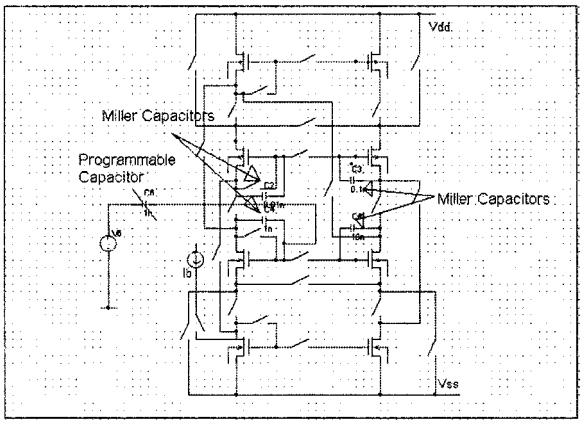
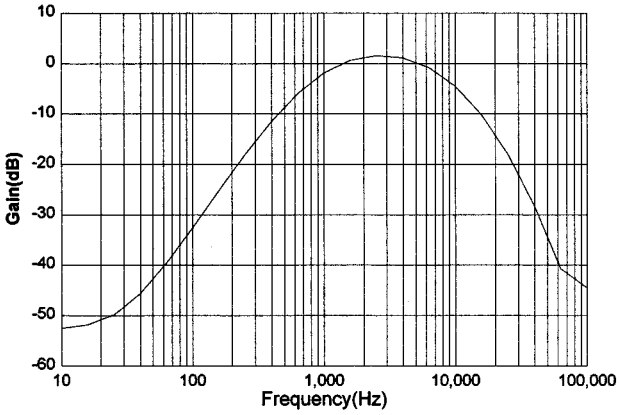


Fig. 6. (a) Circuits obtained by evolution; their design is unusual for common practice, thin dotted lines show parasitic paths corresponding to open switches and (b) best circuit responses in a simulated evolution.



(a)



(b)

Fig. 7. Central cell model with one programmable capacitance and four Miller capacitors; and the evolved bandpass filter response using four cells.

This result compares very well with the one reported in [13], where the basic elements for evolution were bipolar transistors and resistors. While using the FPTA, only 900 evaluations (30 individuals along 30 generations) were necessary, around 10^4 evaluations were needed to obtain a similar dc transfer function in [13]. In addition, the evolved circuit shown above can be directly implemented in a CMOS reconfigurable chip. This result shows that, due to the FPTA architecture, amplifiers can be quickly synthesized through evolutionary means.

2) Gaussian: The following experiment illustrates the evolutionary synthesis of a computational circuit. The goal of evolution was to synthesize a circuit that exhibits a Gaussian voltage-input current-output characteristic. In a previous experiment [14] the circuit topology was fixed and the search/optimization addressed transistor parameters (channel length and width); such evolution proved quite simple. In the FPTA case, the transistor parameters were kept fixed and the search was performed for the 24 binary parameters characterizing switch status. Successful evolution was achieved in multiple runs with populations between 50 and 512, evolving for 50 or 100 generations. The execution time depends on the above variables and on the number of processors used (usually 64 out of the 128 available), averaging around 20 min (the same evolutions took about 2 days on a SUN SPARC 10). In some runs the human designed circuit [14] was rediscovered by evolution. It is interesting to analyze in more detail the unusual solutions found by evolution. Circuits like those illustrated in Fig. 6(a) resulted from evolutionary synthesis and are very similar to the human designed solution. Thin dotted lines illustrate parasitic paths through open switches. Thicker dotted lines show connections that existed in the human-designed circuit but are missing in the circuits in Fig. 6(a). Fig. 6(b) shows the dc transfer function of some circuits that resulted from this experiment: the first two responses correspond to the circuits in (a). Some other responses from the same generation are illustrated for comparison. As it is easy to observe, these circuits are outside normal design practices, e.g., the transistors P2, P4, and N8 (see Fig. 3) on the left circuit in Fig. 6(a) appears to have floating gates. The reality is that the switches have a big, but finite, resistance in the Off state (∼MOhms/GOhms) and a nonzero resistance/impedance in the On state (∼tens/hundreds of Ohms). (Interestingly, this can protect the circuit by limiting the maximum current that can flow when certain closed switches make a path between Vdd and Gnd).

The same evolutionary experiment was performed in hardware on test chips. Four chips were programmed in parallel with bit-string configurations corresponding to four individuals out of a population of 100. As in simulation, evolution led to "Gaussian" circuit solutions within 200–300 generations in approximately 4 min.

3) Filters: The following illustrates the extrinsic evolution, using the central cell model, of a bandpass filter with a passing band between 1 and 10 kHz. Fig. 7 depicts the cell model used to evolve this filter, with a programmable input capacitance and four fixed Miller capacitors. The programmable capacitance can assume eight different values, according to the contents of a particular region of the bitstring. Four interconnected central cells were used in this experiment. This same figure depicts the evolved circuit response, where we can observe that the circuit achieved a 40-dB/decade roll-off outside the passing band. Interestingly, although no large capacitor values were used (they ranged from 10pF to 10nF), evolution was able to determine a circuit solution that exhibited a low-frequency behavior.

VI. Discussions

The aim of those experiments was to test the concept of EHW using a platform specifically designed as a EORA. The portability of the evolved circuits has also been of interest as described in this section.

A. Portability Problem

Previous literature remarks that solutions obtained by evolutionary design suffer from what will be referred to in the following as the portability problem. For example, it was observed in [13] that some circuits obtained through evolutionary design on one HW platform had a different behavior when tested on a second platform, although the two were of similar type/construction. This is strongly related to differences in the set of characteristics that evolution exploited in one platform and can not exploit in the second. Particularly, evolution can explore subtle properties of the silicon, and parasitic effects, which vary even between "identical" chips. A similar situation may be encountered when attempting to port to HW the result of a solution evolved in SW. This is usually caused by limitations in the simulator, such as the accuracy of the circuit model, incomplete information of the fabrication process, convergence problems, initial circuit conditions, etc. Finally, a third portability problem is the one of validating in SW a circuit evolved in HW. In addition to the problems in simulation accuracy, this problem may also be related to the failure to model, in simulation, certain conditions of the hardware experiments, such as output loads and the initial charge of parasitic capacitances.

B. Mixtrinsic Evolution

To solve this portability problem, we developed a third approach to EHW, called mixtrinsic evolution (ME), presented in detail in [15]. Mixtrinsic EHW encompasses a family of techniques for a variety of ways of combining intrinsic and extrinsic modes. The most straightforward alternative is to evaluate each individual both in hardware and in software and assign it a combined fitness function, e.g., an average of the individual fitness for hardware and software evaluation. This constrains evolution to a solution that jointly simulates well in SW, and performs well in HW, i.e., a solution that relies on and exploits only the HW characteristics included in the SW model for producing the desired behavior. The achieved solutions are robust, more likely to be in agreement with common design rules and, if novel, more likely to be patentable (i.e., to have generality and not depend on a fabrication process).

VII. Conclusion

Evolvable hardware requires evolution-oriented architectures beyond those currently implemented in COTS reconfigurable hardware. An FPTA architecture is proposed as a platform for evolutionary experiments. The FPTA allows reconfiguration at transistor level, allows implementation of both analog and digital circuits, offers a variable level of reconfiguration granularity, allows gradual/(gray-level) control of internal switches, has a transparent architecture, and supports any control bitstring configuration without the danger of damaging the chip [3], [16]. The FPTA is flexible and can map a variety of known subcircuits. Experiments with the FPTA bring further evidence on the possibility of using evolutionary algorithms for automatic synthesis of electronic circuits [16] and demonstrate the capability of on-chip automatic synthesis/reconfiguration [18] to achieve newly imposed functionality.

References

[1] J. R. Koza, F. H. Bennett III, D. Andre, and M. A. Keane, *Genetic Programming III—Darwinian Invention and Problem Solving*. San Francisco: Morgan Kaufman, 1999.

[2] A. Thompson, "An evolved circuit, intrinsic in silicon, entwined in physics," in *Int. Conf. Evolvable Systems*: (Lecture Notes in Computer Science) Springer-Verlag, 1996, pp. 390–405.

[3] A. Stoica, D. Keymeulen, R. Tawel, C. Salazar-Lazaro, and W. Li, "Evolutionary experiments with a fine-grained reconfigurable architecture for analog and digital CMOS circuits," in *Proc. 1st NASA/DoD Workshop Evolvable Hardware*: IEEE Computer Society Press, 1999, pp. 76–84.

[4] E. Sanchez, *Field Programmable Gate Array (FPGA) Circuits, Toward Evolvable Hardware: The Evolutionary Engineering Approach*, E. Sanchez and M. Tomassini, Eds: Springer-Verlag, 1996, pp. 1–18.

[5] I. Ozsvald, "Short-circuiting the design process: Evolutionary algorithms for circuit design using reconfigurable analog hardware," M.Sc. thesis, COGS, Univ. Sussex, 1998.

[6] Motorola Semiconductor Technical Data. Advance Information Field Programmable Analog Array 20-Cell Version MPAA020, Motorola, Inc., 1997.

[7] M. Sipper, D. Mange, and A. Perez-Uribe, Eds., "Evolvable systems: From biology to hardware," in *Proc. 2nd Int. Conf., ICES 98*. Lausanne, Switzerland: Springer-Verlag LNCS, 1998.

[8] A. Hamilton *et al.*, "PALMO: Field programmable analog and mixed-signal VLSI for evolvable hardware," in *Proc. 2nd Int. Conf. Evolvable Systems*, M. Sipper, D. Mange, and A. Pérez-Uribe, Eds: Springer-Verlag (LNCS), 1998, vol. 1478, pp. 335–344.

[9] P. Layzell, "A new research tool for intrinsic hardware evolution," in *Proc. 2nd Int. Conf. Evolvable Systems*, M. Sipper, D. Mange, and A. Pérez-Uribe, Eds: Springer-Verlag (LNCS), 1998, vol. 1478, pp. 47–56.

[10] Lattice Semiconductor Corporation, *ispPAC Handbook, Programmable Analog Circuits*, Sept. 1999.

[11] R. Zebulum, A. Stoica, and D. Keymeulen, "A flexible model of a CMOS field programmable transistor array targeted for hardware evolution," in *Proc. 3rd Int. Conf. Evolvable Systems: From Biology to Hardware (ICES2000)*, Edinburgh, Apr. 17–19, 2000, pp. 274–283.

[12] D. Levine, "PGA Pack Parallel Algorithm,", http://www.unix.mcs.anl.gov/ ~levine/PGAPACK.

[13] A. Thompson, P. Layzell, and R. Zebulum, "Explorations in design space: Unconventional electronics design through artificial evolution," *IEEE Trans. Evolutionary Computat.*, vol. 3, pp. 167–196, Sept. 1999.

[14] A. Stoica, "On hardware evolvability and levels of granularity," in *Proc. Int. Conf. Intelligent Systems Semiotics 97: A Learning Perspective*. Gaithersburg, MD: NIST, Sept. 22–25, 1997.

[15] A. Stoica, R. Zebulum, and D. Keymeulen, "Mixtrinsic evolution," in *Int. Conf. Evolvable Systems*, Edinburgh, U.K., Apr. 2000, pp. 208–217.

[16] A. Stoica, "Toward evolvable hardware chips: Experiments with a programmable transistor array," in *Proc. 7th Int. Conf. Microelectronics Neural, Fuzzy Bio-Inspired Systems*. Granada, Spain: IEEE Comp. Sci. Press, Apr. 7–9, 1999, pp. 156–162.

[17] T. Higuchi *et al.*, "Real-world applications of analog and digital evolvable hardware," *IEEE Trans. Evolutionary Computation*, vol. 3, pp. 220–235, Sept. 1999.

[18] D. Keymeulen, A. Stoica, and R. Zebulum, "Fault-tolerant evolvable hardware using field programmable transistor arrays," *IEEE Trans. Reliability*, vol. 49, 2000. (Special Section on Fault-Tolerant VLSI Systems).